

```

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#include<graphics.h>
#define SIZE 100

float arr[SIZE][8],arr1[SIZE][8],trial[8];
float F,CR;
float n,x;
int i,j,t_max=100,count,a,b,c;
float avg,sum,sum_sd;
double dev[SIZE],temp,sd;
int store[SIZE];
time_t t1,t2;
int gd=DETECT,gm;
FILE *fptr;

float fn(float x1,float x2,float x3, float x4, float x5, float x6, float x7, float x8)
{
return(x1+x2+pow(x3,2)-x4+x5-pow(x6,3)+(2*x7)+x8+10000);
}

void clear(float arr[SIZE][8])
{
for(i=0;i<SIZE;i++)
{
for(j=0;j<8;j++)
{
arr[i][j]=0;
}
}
}

/*
void graph(void)
{
setbkcolor(WHITE);
setcolor(RED);
line(50,40,50,400);
line(50,400,500,400);
for(i=1;i<=6;i++)
{
line(50+i*450/6,400,50+i*450/6,405);
line(45,400-i*360/6,50,400-i*360/6);
}
outtextxy(50,405,"0");
outtextxy(40,395,"0");
outtextxy(275,410,"x1");
outtextxy(20,220,"x2");
setcolor(CYAN);
circle(50+3*450/6,400-2*360/6,2);
setcolor(MAGENTA);
for(i=0;i<10;i++)
circle(fabs(50+arr[i][0]*450/6),fabs(400-arr[i][1]*360/6),2);
getchar();
cleardevice();
}*/

void rprod(void)
{
for(i=0;i<SIZE;i++)
store[i]=0;
for(i=0;i<SIZE;i++)
{
one: n=random(SIZE);
if(store[n]==1)
goto one;
else
store[n]=1;
do a=random(SIZE);
while(a==i);
do b=random(SIZE);
while(b==a || b==i);
do c=random(SIZE);
}
}

```

```

        while (c==a || c==b || c==i);
    for(j=0;j<8;j++){
        trial[j]=arr[c][j]+F*(arr[a][j]-arr[b][j]);
    }
    /*trial[1]=arr[c][1]+F*(arr[a][1]-arr[b][1]);*/
    for(j=0;j<8;j++){
        {
            x=((float)(rand()%10000))/10000;
            if(x>CR) trial[j]=arr[n][j];
        }
        if(fn(arr[i][0],arr[i][1],arr[i][2],arr[i][3],arr[i][4],arr[i][5],arr[i][6],arr[i][7]) > fn(trial[0],trial[1],trial[2],trial[3],trial[4],trial[5],trial[6],trial[7]))
        {
            for(j=0;j<8;j++){
                arr1[i][j]=trial[j];
            }
        }
        else
        {
            for(j=0;j<8;j++){
                arr1[i][j]=arr[i][j];
            }
        }
    }
}
clear(arr);
for(i=0;i<SIZE;i++){
    {
        for(j=0;j<8;j++){
            arr[i][j]=arr1[i][j];
        }
    }
}
clear(arr1);
}

void main(void)
{
    fptr=fopen("d:\students\gautam\data.dat","w");
    t1 = time(NULL);
    srand((unsigned) time(&t1));
    clear(arr);
    clear(arr1);
    /*initgraph(&gd,&gm,"");*/

    /*GENERATING RANDOM VALUES FOR FIRST GENERATION*/
    /*printf("First generation\n");*/
    for(i=0;i<SIZE;i++){
        {
            for(j=0;j<8;j++){
                n=((float)(rand()%10000))/10000;
                arr[i][j]=(6*n)/0.9999;
                /*n=((float)(rand()%10000))/10000;
                arr[i][1]=(6*n)/0.9999;*/
            }
        }
    }
    /*graph();*/
    sd=0;
    sum_sd=0;
    sum=0;

    F=0.1;
    do{
        CR=0.1;
        do{
            sum=0;
            sum_sd=0;
            sd=0;
            printf("\nF=%f CR=%f",F,CR);
            for(count=0;count<t_max-1;count++)
                {
                    for(i=0;i<SIZE;i++){
                        {
                            for(j=0;j<8;j++){
                                n=((float)(rand()%10000))/10000;
                                arr[i][j]=(6*n)/0.9999;
                            }
                        }
                    }
                }
        }
    }
}

```

```

        /*printf("Generation number %2d\n",count+2);*/
        rprod();
        /*graph();*/
    }
    for(i=1;i<SIZE;i++){
        sum=sum+fn(arr[i][0],arr[i][1],arr[i][2],arr[i][3],arr[i][4],arr[i][5],arr[i][6],
arr[i][7]);
    }
    avg=sum/SIZE;
    /*for(i=1;i<=SIZE;i++){
        dev[i]=fn(arr[i][0],arr[i][1],arr[i][2],arr[i][3],arr[i][4],arr[i][5],arr[i][6],a
rr[i][7])-avg;
        temp=pow(dev[i],2)/SIZE;
        sum_sd=sum_sd+temp;
    }
    sd=pow(sum_sd,0.5);*/
    fprintf(fptr,"%f %f %f %d\n",F,CR,avg,t_max);
    CR=CR+0.1;
}while(CR<1.0);
F=F+0.1;
}while(F<1.0);
/*closegraph();*/
/*t2=time(NULL);
printf("Time elapsed is %ld\n",t2-t1);*/
/*for(i=1;i<SIZE;i++){
printf("\n%f",fn(arr[i][0],arr[i][1],arr[i][2],arr[i][3],arr[i][4],arr[i][5],arr[i][6],arr[i][7])
);
}*/
/*getchar();*/
}

```